NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPTIMAL AIRCRAFT ROUTING IN A CONSTRAINED PATH-DEPENDENT ENVIRONMENT

by

Norbert J. Karczewski III

September 2007

Thesis Co-Advisors:	Johannes O. Royset
	Ralucca M. Gera
Second Reader:	Craig W. Rasmussen

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 2007

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE Optimal Aircraft Routing in a Constrained Path-Dependent Environment

5. FUNDING NUMBERS

6. AUTHOR(S) Norbert J. Karczewski III

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

In this thesis, we present a method of automatically generating a route of flight for an aircraft, or a group of aircraft flying in formation, from an origin to a destination in the presence of threats. The threats encountered at a point of the route are a function of the route used to arrive there. The route is constrained by limits on one or more resources, such as fuel and time, expended over the course of the route. We use a C++ program to implement the method for two scenarios. In the first scenario, we generate optimal routes for a path-dependent radar threat environment. We then compare these results with routes generated for a path-independent radar threat. In a second scenario, we generate a route for a three-dimensional airspace over terrain in the presence of two constraints and multiple threats that vary dependent upon the route taken. The computing time required to generate a route is sufficiently short for use of the method in mission planning tools. Recommendations for future research and model improvement conclude the thesis.

14. SUBJECT TERMS Flight Planning, Routing, Aircraft, Network Models, Shortest Path, Optimization, Radar Threats, Path-Dependence, Lagrangian Relaxation

15. NUMBER OF PAGES
57

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

OPTIMAL AIRCRAFT ROUTING IN A CONSTRAINED PATH-DEPENDENT ENVIRONMENT

Norbert J. Karczewski III
Major, United States Marine Corps
B.S., United States Naval Academy, 1993

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN OPERATIONS RESEARCH
AND
MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
September 2007

Author: Norbert J. Karczewski III

Approved by: Johannes O. Royset
Thesis Co-Advisor

Ralucca M. Gera
Thesis Co-Advisor

Craig W. Rasmussen
Second Reader

James N. Eagle
Chairman, Department of Operations Research

Clyde L. Scandrett
Chairman, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis, we present a method of automatically generating a route of flight for an aircraft, or a group of aircraft flying in formation, from an origin to a destination in the presence of threats. The threats encountered at a point of the route are a function of the route used to arrive there. The route is constrained by limits on one or more resources, such as fuel and time, expended over the course of the route. We use a C++ program to implement the method for two scenarios. In the first scenario, we generate optimal routes for a path-dependent radar threat environment. We then compare these results with routes generated for a path-independent radar threat. In a second scenario, we generate a route for a three-dimensional airspace over terrain in the presence of two constraints and multiple threats that vary dependent upon the route taken. The computing time required to generate a route is sufficiently short for use of the method in mission planning tools. Recommendations for future research and model improvement conclude the thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

In this thesis, we present a method of automatically generating a route of flight for an aircraft, or a group of aircraft flying in formation, from an origin to a destination in the presence of threats. The threats encountered at a later point of the route are a function of the route used to arrive there. The route is constrained by limits on one or more resources, such as time and fuel, expended over the course of the route.

We model the airspace as a directed network in which the nodes of the network represent potential waypoints in three-dimensional airspace, and the arcs represent segments of flight between waypoints. Since the magnitude of the threat at any point is a function of the route used to arrive there, the probability of mission success while transiting along an arc depends on the arcs used in the route. We formulate a Path-dependent Constrained Shortest-Path Problem (PCSP) to find the route of flight with the maximum probability of success.

We solve the PCSP problem in two stages. In the first stage, we relax the PCSP problem by eliminating the path dependency in the objective function and formulate an integer linear program. This integer program is a discrete shortest path problem with side constraints. Through the technique of Lagrangian Relaxation, we incorporate the side constraints into the objective function, resulting in a discrete shortest-path problem. We then solve this shortest-path problem by using a label-correcting shortest-path algorithm. The Lagrangian relaxation solution provides us with a bound on the optimal value of the PCSP. We may also have obtained a feasible path satisfying our side constraints.

The second stage utilizes the solution obtained in the prior stage. In this stage, we enumerate paths from our origin to our destination. In the course of enumeration, we use the true path-dependent probabilities of mission success. We use bounds obtained by feasible solutions and Lagrangian Relaxation to limit the enumeration. Any time we find another solution with a better objective function value, we improve a bound that further limits enumeration. When we have completed the enumeration process, we will have obtained our optimal path.

We then present two scenarios to illustrate the solution strategy. In the first scenario, we compare the optimal routes when modeling Surface-to-Air Missile (SAM) threats in a path-independent manner to optimal routes when modeling SAM threats in a path-dependent manner. In the path-independent model, SAM threats consist of concentric range circles with increasing probability to shoot down an aircraft. In the path-dependent model, the probability to shoot down an aircraft increases at an increasing rate of the amount of time in the SAM range. Optimal routes in the path-independent model, seek to reduce the amount of time in the inner most SAM range circles. In these areas, the probability of shooting an aircraft down is higher than the outer annulus shaped regions. In the path-dependent model, optimal routes seek to reduce the amount of time spent within range of a SAM, regardless of the route location within the range circle. Air intelligence planners now have two different methods modeling radar threats. They can select the method that best fits a particular environment.

In the second scenario, we generate a three-dimensional route for an aircraft from origin to destination in the presence of multiple radar threats and locations where enemy may observe and react to the flight. The enemy may observe aircraft from three locations. Flight within visual range of these locations for an extended period, or flight over more than one of these locations within a given time window will result in a large risk of being shot down by the enemy. Rather than avoiding all locations of enemy activity, planners can determine the optimal route. This route may include flight over some enemy locations. This feature is particularly useful in situations with dense concentrations of enemy activity. The computing time required to generate a route in these scenarios is sufficiently short for use of the method in mission planning tools.

I. INTRODUCTION

A. BACKGROUND

A crucial component in mission planning for aviation units is to determine the route aircraft will take into and out of an objective. While the aircraft traverse the route, they consume limited resources such as time, fuel and ordnance. Additionally, each portion of the route contributes toward or detracts from the probability of mission success. If the enemy observes the flight of aircraft and accurately judges the flight's intent, he can take steps to thwart its mission. These steps include attacking the aircraft to prevent them from reaching their destination, or reinforcing or relocating their targeted objectives. Thus, in developing his plan, the Air Mission Commander (AMC) faces the challenge of selecting the route for each aircraft that maximizes the probability of mission success while consuming no more than the aircraft's allotted resources.

Vital coordination for fire-support and communications and control depend upon integration by orders issued from higher headquarters. Higher headquarters cannot complete and issue their orders until they have received a briefback from subordinate units detailing their plans (FM 90-26, 1990). These integration requirements constrain the amount of time the AMC has to issue orders to aviation units. Thus, planners exploit any automation available in order to plan the flight route as swiftly as possible. The current automation tool employed by Navy and Marine Corps flight planners is Falcon View (FalconView, 2007). This is a flight-planning package allowing overlays of threat information and command and control airspace on maps and charts. However, the route of flight must be manually determined based solely upon the expertise and judgment of the planners. Planners input the route, utilizing Falcon View to generate flight packets containing the route, command and control measures, and other information upon which aircrew rely to navigate their flight.

Unmanned Aerial Vehicles (UAVs) typically accomplish high-risk missions. At present, even the routes of these aircraft are determined based on human judgment and the technical expertise of the UAV controllers. UAV route planners operate under the same urgency as planners for manned flights, since they must also submit their selected route to higher headquarters in order to integrate command and control and fire-support assets. Thus, driven by intense time pressure, UAV mission planners, like AMCs for manned missions, may select routes with a less than optimal probability of mission success.

Planners select routes for manned and unmanned missions based on certain assumptions and experience in order to speed their decision-making. In relying on experience, planners may fall prey to the error of choosing a route used in a successful previous mission. In so doing, they fail to take into account the fact that they are up against an intelligent adversary who also may be planning their course of action based upon previously flown routes. Repeated flights along the same route will unnecessarily expose the aircraft to the threat of ambush by the enemy, due to the predictability of the mission’s route. Planners might attempt to mitigate the threat of enemy ambush by entirely avoiding a previously flown route, but in so doing, expose themselves to other types of threats. If a planning tool were available which would enable planners to quantify the risk of flying over only portions of a route previously flown, they could generate a route optimally balancing the threat from predictability with other threats.

Additionally, planners may assume that conditions at a point along the route will not change regardless of how the flight arrived at that point. Again, this assumption does not take in to account conditions that change resulting from the flight itself. Changes in threat might result from different environmental conditions, such as weather or lighting conditions, which depend on the flight’s arrival time at this point. As an example, if the moon is low on the horizon, one side of a prominent terrain feature might be well illuminated. On the illuminated side, the threat of striking an obstacle or a pilot error due to high workload would be much lower than the side in shadow. As time progresses, the lunar position in the sky changes and might reverse these environmental threats. Hence, routes that differ in time of arrival to a certain point will face different conditions at that point (see Figure 1).

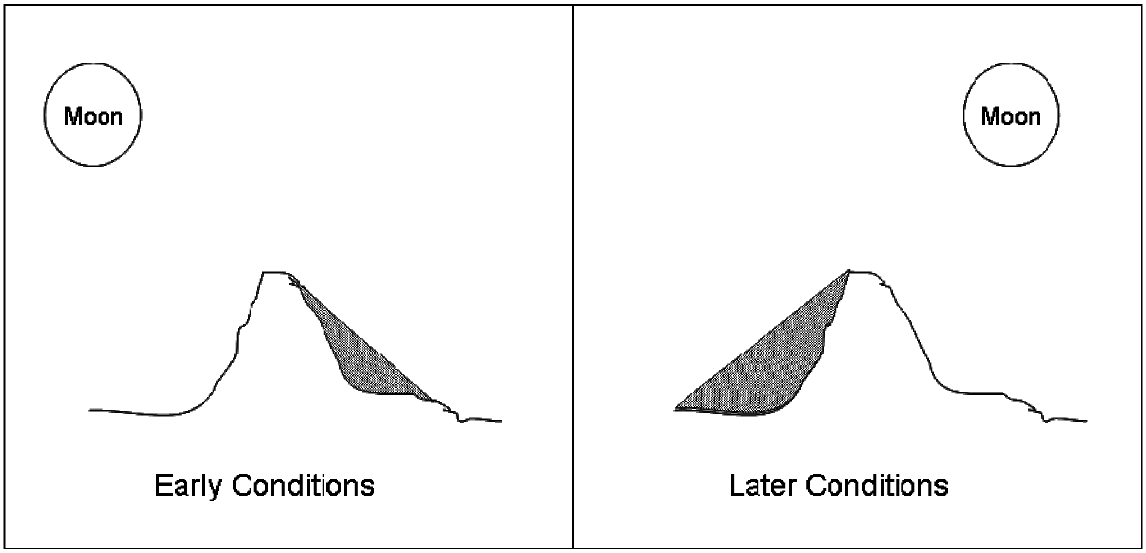

Figure 1. Changing Moon Position Leaves Different Areas in Shadow

Threats from enemy radars at a particular point may also depend on the route taken to that point. For example, the radar threat might increase the longer an aircraft remains within range of the radar. Thus, missiles guided by the radar will shoot the aircraft down with a lower probability when it first enters radar range than after tracking the aircraft for some time. In this situation, the route of flight to a particular point within the radar range has a large effect on the magnitude of threat encountered at that point.

In the preceding examples, future threats depend upon the earlier choice of route. Planners know these threats and their dependence on the selected route, but due to the limited amount of time they have to plan the route, they cannot fully utilize this information in flight planning.

Current mission planning software in use by Navy and Marine Corps aviation units does not incorporate an automatic route generator. There are several automatic route generators in existence, such as CLOAR (CLOAR, 2007), OPUS (OPUS, 2007), and JRAPS, but all of them have a number of modeling and computational shortcomings thoroughly discussed by Carlyle, Royset and Wood (2007a). Additionally, none of these automatic route generators account for threat dependencies.

In this thesis, we present a method of automatically generating a route of flight for an aircraft, or a group of aircraft flying in formation, from an origin to a destination in the presence of threats. The threats encountered at a point of the route are a function of the route used to arrive there. The route is also constrained by limits on one or more resources expended over the course of the route.

B. LITERATURE SURVEY

An automatic route generator for path-dependent environments must address two types of problems handled separately in previous studies. The first type of problem consists of finding the resource constrained shortest path between two points. In this case, a path is feasible only if travel along a path consumes no more resources than allowed by specific capacities. We call this the constrained shortest path problem. The second type of problem is concerned with finding an optimal route when the threats at a point depend upon the sub-path to that point. This is the *path-dependent* shortest-path problem.

1. Techniques to Solve Continuous and Discrete Constrained Shortest-Path Problems

Current studies employ three primary methods of solving the constrained shortest-path problem. These methods are calculus of variations, Nonlinear Trajectory Generation, and network models. Calculus of variations and Nonlinear Trajectory Generation are continuous models describing the trajectory of the aircraft at any given moment in time. Network models represent the airspace by a set of discrete points between which an aircraft may fly.

a. Calculus of Variations

Zabarankin, Uryasev, and Murphy (2006) utilize the techniques of calculus of variations in order to find the minimum risk route of an aircraft through an area containing one radar threat. This technique provides a continuous solution to the problem. Continuous models have an equal or better probability of mission success than network models. The higher probability of mission success results from the fact that the aircraft may follow a curved trajectory, rather than flying directly from point to point as it is restricted to do in a network model. In practice, however, aircraft navigate from point to point in three-dimensional space, either over prominent terrain features at set altitudes, or between points in the airspace identified through navigational aids such as the Global Positioning System (GPS). In order to follow a route determined through the calculus of variations, an aircraft might be required to make several quick turns in succession. The demands on the aircrew to adhere to this route might increase their chance of missing other crucial aspects of the mission, resulting in reduced probability of mission success. A second drawback to the calculus of variations method of Zabarankin et al. (2006) is the fact that the route is determined in the presence of only one threat to mission success. It appears difficult to extend the calculus of variations method to address multiple threats, let alone extend it further to address the path-dependent shortest path problem.

b. Nonlinear Trajectory Generation

Nonlinear Trajectory Generation (NTG) is a software package developed to solve optimal trajectory generation problems for constrained systems (Milam, 2003). NTG uses the NPSOL (NPSOL 5.0, 2007) solver as a nonlinear optimization tool in order to provide a continuous solution. Misovec, Inanc, Wohletz and Murray (2003) and Inanc, Misovec and Murray (2004) use NTG to determine a route for a UAV with maximum probability of success in the presence of multiple threats. The main advantage of their work is the speed with which NTG arrives at a solution. This speed comes at the cost of sensitivity to an initial route selection. Due to this sensitivity, the solutions arrived at through this method are guaranteed to be locally optimal, but may not be globally optimal. Thus, NTG is best suited for dynamic situations where the task is to modify a pre-selected route, rather than a planning tool to find the optimal route.

c. Network Models

In using a network model to solve the discrete constrained shortest-path problem, Zabarankin et al. (2006) use a label-setting algorithm to find the optimal route.

However, in their study, only one constraint on resources and one threat to mission success is considered. Carlyle, Royset and Wood (2007b) handle one and two constraints on resources, as well as multiple threat sources by using the technique of Lagrangian Relaxed Enumeration (LRE) to find an optimal solution. We will build upon LRE in this thesis since it is competitive with the label-setting algorithm and readily extensible to handle path-dependent situations.

2. Studies of Path-Dependent Shortest Path Problems

Tan and Leong (2004) show that path-dependent discrete shortest path problems are NP-complete, noting an exponential amount of storage space may be required in order to describe the dependency. Orda and Rom (1990) propose several algorithms for handling path-dependent discrete shortest-path problems in communications and data transmission applications. These studies only seek to minimize total arc length in a path through the network and do not consider consumption of resources along this path. Thus, their approaches do not readily extend to cases with side constraints on such resources.

Other studies, such as Davies and Lingras (2003) and Fan, Kalaba and Moore (2005), deal with dynamic planning of routes in networks where the path through the network affects travel costs between nodes. In this thesis, we consider preplanning of routes in their entirety prior to mission departure. The methods of Davies et al. (2003) and Fan et al. (2005) are for situations in which a route commences before obtaining all pertinent information. They also do not consider side constraints that limit route choice. Thus, their techniques are difficult to extend to the aircraft routing problem under investigation.

C. ORGANIZATION

The remainder of this thesis formulates and solves a model of a discrete path-dependent shortest-path problem for aircraft routing, presents solution strategies and optimal routes to different path-dependent scenarios. Chapter II presents the mathematical formulation of the model and solutions strategies for solving it. Chapter III compares results using a path-independent model and a path-dependent model. Chapter IV presents a three dimensional path-dependent scenario. Chapter V concludes the thesis and suggests areas for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MODEL FORMULATION AND SOLUTION STRATEGY

A. AIRCRAFT ROUTING MODEL

The airspace is modeled as a directed network $G = \left(N, E\right)$ in which the nodes of the network $n \in N$ represent potential waypoints in three-dimensional airspace, and the arcs $(i, j) \in E$ represent segments of flight between waypoints, from node i to node j, where $i, j \in N$. The aircraft's entry point into the modeled airspace is labeled as node $s \in N$ and its destination by the node $t \in N$. We define an s-t path, $P(s,t)$, to be a sequence of nodes from s to t, $s = i_o, i_1, i_2, ..., i_k = t$. This sequence of nodes represents a series of arcs used to travel from one node to the next in the sense that for any two consecutive nodes i_y and i_{y+1} in the path, $(i_y, i_{y+1}) \in E$. We denote the set of all paths from s to t $\mathcal{P}\left(s,t\right)$. Define a sub-path of $P\left(s,t\right)$ that starts at s by $P(s,i): s = i_o, i_1, i_2, ..., i_m = i$, where m is a natural number with $m < k$. We indicate an arc $(i_y, i_{y+1}) \in E$ shows up on a path by the symbol $(i_y, i_{y+1}) \in P\left(s,t\right)$. If no nodes are repeated in the (sub)path, we then call it a simple (sub)path.

A path represents a route for the aircraft between s and t. Current doctrine views revisiting waypoints in a route as unattractive due to the increased likelihood of drawing enemy fire. Since the nodes of the network represent waypoints for an aircraft route of flight, a path that takes in to account this doctrinal preference would not repeat any nodes. Thus, in this thesis we only consider simple paths and sub-paths. In principle, the methods in this thesis could handle non-simple paths, but to do so may cause weakening of bounds in path-enumeration and is beyond the scope of this thesis.

Each arc $(i, j) \in E$ has data associated with it. The primary data is the probability of success while transiting this arc. This probability of mission success is a function of the sub-path taken to arrive there and the arc itself:

$$q_{(i,j)} = \phi\left(P(s,i), \left(i, j\right)\right) \tag{1}$$

Secondary data for each arc, denoted weights, represent additive metrics such as fuel, time, and ordnance. Let K be the total number of secondary weights for an arc. Then, for a given arc, $(i, j) \in E$, $f^1_{(i,j)}, f^2_{(i,j),} ..., f^K_{(i,j)}$ represent these weights. The aircraft has a limited capacity for each of these weights, for instance representing the total amount of fuel, mission time and ordnance available. We let g_i be the upper limit on the i-th weight along a path. We obtain an optimal route, $P^*(s,t) \in \mathcal{P}(s,t)$, by solving the following optimization problem:

$$\zeta = \max_{P(s,t)\in\mathcal{P}(s,t)} \prod_{(i,j)\in P(s,t)} q_{(i,j)} \qquad \text{Objective Function (2)}$$

$$\text{subject to} \sum_{(i,j)\in P(s,t)} f^k_{(i,j)} \le g_k, \ \forall\ k = 1,2,...,K \qquad \text{Side Constraints (3)}$$

In order to compare results from this optimization problem to future relaxations, we now perform a logarithmic transformation to our objective function. Since the logarithm function is monotonically increasing, we can equivalently consider the problem:

$$\zeta' = \max_{P(s,t)\in\mathcal{P}(s,t)} \ln\left(\prod_{(i,j)\in P(s,t)} q_{(i,j)} \right) \qquad \text{Objective Function} \quad (4)$$

$$\text{subject to} \sum_{(i,j)\in P(s,t)} f^k_{(i,j)} \le g_k, \ \forall\ k = 1,2,...,K \qquad \text{Side Constraints (5)}$$

The values of (4) are all non-positive. Equivalently, we opt to minimize the negative value of the objective function in (4). We then obtain the Path-dependent Constrained Shortest-path Problem (PCSP) defined as follows:

$$\zeta^* = \min_{P(s,t)\in\mathcal{P}(s,t)} \sum_{(i,j)\in P(s,t)} -\ln\left(q_{(i,j)}\right) \qquad \text{Objective Function (6)}$$

$$\text{subject to} \sum_{(i,j)\in P(s,t)} f^k_{(i,j)} \le g_k, \ \forall\ k = 1,2,...,K \qquad \text{Side Constraints (7)}$$

where ζ^* is the logarithmically-transformed optimal probability of mission success. We note the optimal value of (2-3) relates to the optimal value of (6-7) in the following manner:

$$\zeta = \exp\left(-\zeta^*\right) \qquad (8)$$

We now present our solution strategy to solve the PCSP.

B. SOLUTION STRATEGY

We solve the PCSP problem in two stages. In the first stage, we relax the PCSP problem by eliminating the path dependency in the objective function and formulate an integer linear program. This integer program is a discrete shortest path problem with side constraints. Through the technique of Lagrangian Relaxation (Ahuja, Magnanti & Orlin, 1993, pp. 598-638), we incorporate the side constraints into the objective function, resulting in a discrete shortest-path problem. We then solve this shortest-path problem using a label-correcting shortest-path algorithm. The Lagrangian relaxation solution provides a lower bound on the optimal value of the PCSP. We may also have obtained a feasible path satisfying our side constraints.

The second stage utilizes the solution obtained in the prior stage. In this stage, we enumerate paths from the origin to the destination. In the course of enumeration, we use the true path-dependent mission success probabilities. Enumeration of feasible paths that do not violate our side constraints is limited by using the upper bound obtained by a feasible solution and a lower bound obtained from the Lagrangian Relaxed solution. Any time we find another feasible solution with a better objective function value, we update the upper bound and use this new value to limit our enumeration. When we have completed the enumeration process, we will have obtained the optimal path.

1. Integer Program Formulation and Lagrangian Relaxation

The first step in stage one is to define of "best-case" risks. We eliminate the path dependency by assigning the minimum value of all possible "risks" that can be experienced at an arc. Thus, for a given arc, $(i, j) \in E$, we define a risk-unit as:

$$c_{(i,j)} = \min_{P(s,i)\in\mathcal{P}(s,i)} \left[-\ln\left(q_{(i,j)}\right)\right] \tag{9}$$

This risk-unit is path-independent, corresponding to the most favorable probability of success when transiting arc (i, j) that can possibly occur.

We now follow the method of Carlyle et al. (2007b), assigning an order to all arcs $(i, j) \in E$, labeling them $e_1, ..., e_{|E|}$, and designating a vector of risk units $\mathbf{c} = \left(c_{e_1}, ..., c_{e_{|E|}}\right)$. We let $\mathbf{x}$ be a binary vector with dimension $|E| \times 1$. An entry of 1 in $\mathbf{x}$ signifies a particular arc will be traversed, whereas an entry of zero signifies it will not. Let A be an $|N| \times |E|$ incidence matrix for G such that if $e = (i, j) \in E$, then $A_{ie} = 1$, $A_{je} = -1$, and $A_{ue} = 0$ for all $u \in N \setminus \{i, j\}$. Form an $|N| \times 1$ balance of flow vector $\mathbf{b}$ such that $b_s = 1$, $b_t = -1$, and $b_i = 0$ for all $i \in N \setminus \{s, t\}$. Now for each constraint, we collect the constraint weights, $f_{(i,j)}^k$, into a row vector, $\mathbf{f}_{(i,j)}$, where K designates the total number of constraints. We then collect these row vectors into a $K \times |E|$ matrix F where each row corresponds to one of the constraints. Finally, we let $\mathbf{g} = [g_1, ..., g_K]^T$ be the vector of constraint limits.

We now utilize this data and nomenclature in order to formulate the path-independent integer linear program required for stage one of our solution strategy. This discrete constrained shortest-path problem (CSPP) is:

Sets:

N:	Nodes
E:	Arcs

Parameters:

A:	incidence matrix
F:	constraint matrix
$\mathbf{b}$:	balance of flow vector
$\mathbf{g}$:	constraint limit vector
$\mathbf{c}$:	risk-unit vector

Decision variable:

$\mathbf{x} \in \{0,1\}^{|E|}$

Constraints and objective function:

$$z = \min_{\mathbf{x}} \mathbf{cx} \qquad \text{Objective function (10)}$$

$$\text{subject to} \quad A\mathbf{x} = \mathbf{b} \qquad \text{Balance of Flow Constraints (11)}$$

$$F\mathbf{x} \leq \mathbf{g} \qquad [\boldsymbol{\lambda}] \qquad \text{Side Constraints (12)}$$

Any binary vector $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{b}$ uniquely defines a path from s to t. Since $c_{(i,j)} \geq 0$ for all $(i, j) \in E$ there always exists an $\mathbf{x}$ defining a simple path.

In the final step of the first stage, the Lagrange multiplier $\boldsymbol{\lambda}$, a $1 \times K$ vector, is used to relax the CSPP by incorporating the constraints into the objective function. The resulting Lagrangian-relaxed program is as follows:

$$z(\lambda) = \min_{\mathbf{x}} \mathbf{cx} + \boldsymbol{\lambda}(F\mathbf{x} - \mathbf{g}) \qquad \text{Objective function (13)}$$

$$\text{subject to } A\mathbf{x} = \mathbf{b} \qquad \text{Balance of Flow Constraints (14)}$$

For any fixed $\boldsymbol{\lambda}$, the Lagrangian-relaxed program, consisting of (13) and (14), is a shortest-path problem, referred to as LRSPP, which can be solved quickly (Ahuja et al., 1993, pp. 598-638). The optimal solution obtained, $\mathbf{x}_{LB}$, provides a lower bound $\underline{z} = \mathbf{cx}_{LB}$ for the PCSP. Seeking to find the greatest lower bound possible, we solve the following Lagrange multiplier optimization problem:

$$\underline{z}^* = \max_{\boldsymbol{\lambda} \geq 0} \min_{\mathbf{x}} \mathbf{cx} + \boldsymbol{\lambda}(F\mathbf{x} - \mathbf{g}) \qquad \text{Objective function (15)}$$

$$\text{subject to } A\mathbf{x} = \mathbf{b} \qquad \text{Balance of Flow Constraints (16)}$$

The optimal solution, $\underline{z}^*$, to this program is the greatest lower bound for our LRSPP and PCSP. In the course of finding this lower bound, we may have found a feasible solution $\hat{\mathbf{x}}$ that satisfies our side constraints.

If $\hat{\mathbf{x}}$ is a feasible solution that satisfies our side constraints (12), it represents a corresponding unique simple path, $\widehat{P}(s,t)$, from s to t. We now compute an upper bound, $\overline{z}$, for our PCSP using the true path-dependent probabilities of success along an arc:

$$\overline{z} = \sum\nolimits_{(i,j) \in \hat{P}(s,t)} -\ln q_{(i,j)} \qquad (17)$$

If no feasible solution has been located, we can still calculate a weak upper bound. In order to do so, we must determine the maximum number of arcs that can be traversed in a path from s to t. In order to determine this number, for all k, $1 \le k \le K$, we let $\eta_k = g_k / f_{\min}^k$ where $f_{\min}^k \equiv \min\left\{ f_e^k \mid e \in E \right\}$. The maximum number of arcs which are traversed is $\eta_{\max} \equiv \max\left\{ \eta_k \mid 1 \le k \le K \right\}$. Our PCSP upper bound then becomes, $\overline{z} = \eta_{\max} c_{\max}$, where and $c_{\max} \equiv \max\left\{ c_e \mid e \in E \right\}$.

2. Enumeration

The next stage is to begin enumerating possible paths $P(s,t) \in \mathcal{P}(s,t)$ by creating a sub-path, which we hope to extend through the addition of arcs into a path from s to t. We cease extending this sub-path, if we can project that we will violate the side constraints or we will not improve our current best solution. If we find a feasible path which improves our value for $\overline{z}$, then we update our upper bound and retain that particular path as our candidate for our optimal solution. When we have exhausted this process, the best path we have found during enumeration is the optimal solution to the PCSP problem.

In the process, we keep track of several values for our current sub-path. These values are R, L and G_k where $k = 1, 2, ..., K$ representing the total risk, Lagrangian-relaxed risk, and total amount of resource k we have consumed on our current sub-path, respectively. We define the Lagrangian-relaxed risk as the contribution to the value of $\underline{z}^*$ in (14) from transiting along an arc, $\mathbf{c}' \equiv \mathbf{c} + \boldsymbol{\lambda} F$. To begin enumeration we set the value of L to $-\boldsymbol{\lambda}\mathbf{g}$ and all other values to zero.

Now, for all nodes $i \in N \setminus \left\{ t \right\}$, we calculate the minimum quantities required to travel from i to t. The quantities $r(i)$, $l(i)$, and $d_k(i)$ denote the minimum amount of risk units, Lagrangian-relaxed risk units, and resource k, for all $1 \le k \le K$, required in all possible paths from node i to node t. This process is simply $K+2$ shortest-path calculations carried out backwards from t. Now we use a Last-In-First-Out data structure called a stack to keep track of our sub-path. Beginning at our origin node s, we consider an arc $(i, j) \in E$ departing our current node, $i \in N$. First, we label this arc as "seen." We determine if we have met the following four conditions:

1. We cannot have previously been through node j in our current sub-path.

2. $R - \ln\left(q_{(i,j)}\right) + r(j) \le \overline{z}$, where $q_{(i,j)} = \phi\left(P(s,i),(i,j)\right)$ and $P(s,i)$ consists of the arcs in the sub-path.

3. $L - \ln\left(q_{(i,j)}\right) + \boldsymbol{\lambda}\mathbf{f}_{(i,j)} + l(j) \le \overline{z}$

4. $G_k + f_{(i,j)}^k + d_k(j) \le g_k$ for all arc weights $k = 1, 2, ..., K$.

If these conditions are satisfied, we extend our sub-path to node j by placing arc (i, j) on our stack. We update our path values by adding the risk unit and side constraints to the previous values. If $j = t$, we update $\overline{z} = R$. If the conditions are not satisfied, or $j = t$, we now repeat the process considering the next arc leaving from our current node which we have not previously labeled as "seen."

When we have labeled all the arcs leaving our current node as "seen," we step back to the previous node. In order to accomplish this we backtrack to the second to last node on our sub-path. To accomplish the backtracking we remove the top arc, (i, j) from our stack. We update our data as follows:

1. $R = R + \ln\left(q_{(i,j)}\right)$

2. $L = L + \ln\left(q_{(i,j)}\right) - \boldsymbol{\lambda}\mathbf{f}_{(i,j)}$

3. $G_k = G_k - f_{(i,j)}^k$ for all arc weights $k = 1, 2, ..., K$

4. We remove our "seen" label from all arcs leaving node j.

Our enumeration will end once we have backtracked to our origin node s and we have previously labeled all departing arcs at s as "seen." At this moment, we will have obtained the optimal path from s to t for PCSP. We can also choose to terminate our enumeration early by allowing a certain tolerance, ε. When we have obtained a path from s to t which results in a corresponding total risk value, R, that is within this tolerance percentage of our lower bound obtained through Lagrangian relaxation, $R - \underline{z} < \varepsilon$, we cease enumeration.

3. Preprocessing

We can reduce the size of the network by utilizing the preprocessing techniques outlined by Dumitrescu and Boland (2003) and Carlyle et al (2007b). These techniques improve upon those proposed by Aneja, Aggarwal, and Nair (1983), identifying arcs in the network that never lie on an optimal path and removing these arcs from the network. To determine if a given arc, $(i, j) \in E$, could be deleted from the network, for a given side constraint g_k, where $1 \le k \le K$, we compute the minimum amount of constrained resource required to arrive at node i from node s, $D_k(i)$. We have already computed the minimum amount of resource required to proceed from node j to node t, $d_k(j)$. If $D_k(i) + f_{(i,j)}^k + d_k(j) > g_k$, (i, j) cannot be on a feasible path and we delete (i, j) from the network. We repeat this process with all arcs and all side constraints until no more arcs can be deleted. When we have obtained an upper bound, $\overline{z}$, we also apply this procedure utilizing the values of our risk-unit vector $\mathbf{c}$ instead of our constrained resource and replacing "$> g_k$" with "$> \overline{z}$" in our comparison equation. This process will be repeated several times because removal of some arcs may alter the values for $D_k(i)$ and $d_k(i)$ for a node $i \in N$.

As we proceed with enumeration, we can use improvements in the value of our upper bound in additional rounds of preprocessing. Should the value of our upper bound decrease by a predetermined percentage, δ, over the course of enumeration, we can repeat the preprocessing procedure in an attempt to further reduce enumeration. In order to accomplish multiple preprocessing, we store our initial upper bound value as, $\overline{z}_{initial}$.

After performing an update to our upper bound value, if $\overline{z} \leq \delta \cdot \overline{z}_{initial}$, we preprocess again by performing the following steps:

1. Compute the minimum total risk units as defined in (9) required to arrive at all nodes $i \in N \setminus \{s\}$ from node s, $M(i)$.

2. $\overline{z}_{initial} = \overline{z}$

3. For all $(i, j) \in E$, if $M(i) + c_{(i,j)} + r(j) > \overline{z}$, remove (i, j) from the network.

Repeat these three steps until no more arcs can be deleted.

Now, for all nodes $i \in N \setminus \{t\}$, we recalculate the minimum quantities $r(i)$, $l(i)$, and $d_k(i)$, for all $1 \leq k \leq K$, required in all possible paths from node i to node t. Now, we continue enumeration as described above.

THIS PAGE INTENTIONALLY LEFT BLANK

III. COMPARISON OF F/A-18 STRIKE MODELS

This scenario considers routing an F/A-18 strike aircraft at a level altitude between two points in the presence of multiple radar threats. Carlyle et al. (2007a) present a similar scenario under the assumption that the radar threats are independent of the path taken. As we will see in this chapter, modeling the radar threat as an increasing function of time that increases at a rate proportional to the amount of time the aircraft is within the radar's detection range results in differing optimal routes.

A. PATH-INDEPENDENT MODEL

In the path-independent model, Carlyle et al. (2007a) formulate a constrained shortest-path problem on a network consisting of a highly connected grid of nodes. They model the airspace under the assumption that the aircraft will travel at a level altitude. Thus, a 26×38 grid of nodes with a spacing of eight nautical miles (nm) represents a 200 nm by 296 nm airspace. They generate arcs from a node to all other nodes between 16 nm and 120 nm, inclusive, in distance. They use the Euclidean distance of the arc as a surrogate for fuel consumed. Fuel consumption is the only constraint on route selection.

The sources of risk in the model are 15 surface-to-air missile (SAM) sites. They surround each SAM with two or three concentric circles of differing radii (see Figure 3). Aircraft will be shot down at a certain risk per unit of time spent traversing each region. The area of highest risk is the center circle, and the threat decreases step-wise in each larger annulus-shaped region. For a given arc, $e \in E$, the probability of successfully traversing the arc is denoted q_e. As defined previously, the risk unit for a given arc then becomes $c_e = -\ln(q_e)$.

B. PATH-DEPENDENT MODEL

In the path-dependent model, we construct the network in the same fashion with the SAM threats in the same location (see Figure 4). Let $\sigma \in \{1,...,15\}$ represent one of the fifteen different SAMs of Carlyle et al. (2007a). For a given SAM, σ, we assign a parameter, ρ_σ, representing the capability of that SAM to shoot down an aircraft. Let $\psi(\tau_\sigma)$ be an increasing function of the amount of time, τ_σ, an aircraft is within the range of a SAM σ, such that the second derivative of $\psi(\tau_\sigma)$ with respect to τ_σ is greater than zero. An aircraft will be shot down by SAM σ with probability $\rho_\sigma \psi(\tau_\sigma)$. This leads to the following expression for the logarithmically transformed probability of mission success for a route as:

$$\sum_{\sigma=1}^{15} -\ln\left(\left(1-\rho_\sigma \psi(\tau_\sigma)\right)\right) \tag{18}$$

In the first phase of our solution strategy, we minimize the risk unit for a given arc $(i, j) \in E$ by assigning the minimum possible time an aircraft will be in range of each SAM. For example, if node i lies within the range of SAM $\sigma = 1$, we assume the aircraft flew along the radius intersecting node i in order to minimize τ_1. Thus, as depicted in Figure 2, τ_1 would be the time required to travel from v to i plus the time required to travel from i to w.

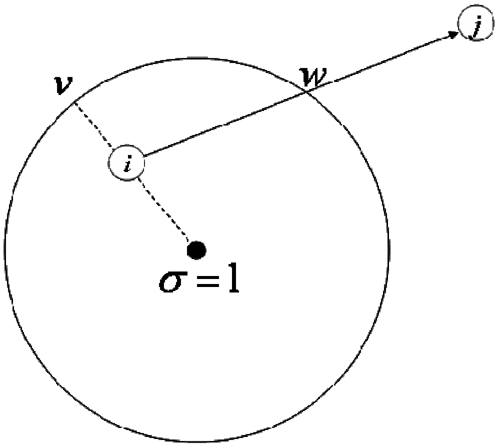

Figure 2. Assigning a value for the minimum time within a SAM

In our CSPP, $c_{(i,j)} = -\ln\left(1-\rho_1 \psi(\tau_1)\right)$. During the enumeration phase, the true amount of time the path spends within each SAM is determined to calculate the path-dependent risk units.

C. COMPARISON OF RESULTS

In this comparison, we set the optimality tolerance to zero in order to compare the optimal routes between models. We ran both models on a Dell Precision PWS690 Intel® Xeon™ CPU 3.37GHz processor, with 3.00 GB of RAM, with programs written and compiled using Microsoft Visual C++ Version 6.0.

Constraint Fuel Units	Independent		Path-Dependent	
	Time (sec.)	Total Risk (Risk Units)	Time (sec.)	Total Risk (Risk Units)
300	0.063	0.265711	0.500	0.536868
305	0.063	0.243143	1.078	0.523524
310	0.110	0.235569	0.781	0.501592
315	0.110	0.215513	0.891	0.488551
320	0.141	0.176352	1.265	0.327882
325	0.078	0.115739	0.484	0.214784
330	0.078	0.098248	0.515	0.206776
335	0.079	0.096505	0.531	0.200152
340	0.094	0.095511	0.515	0.195164
345	0.078	0.085177	0.594	0.191054
350	0.079	0.083495	0.437	0.169694

Table 1. Comparison of Different Model Outputs

Table 1 compares the model run times in seconds and the total risk measured in units between the two models. In both models, the total amount of risk units decreases monotonically as the amount of fuel units increases. Having more fuel enables the aircraft to maneuver the route in order to minimize the risk. The run times for the path-dependent model are always longer than the path-independent model. The longer run times are due to the necessity of recalculating probabilities when we add arcs to a sub-path. Each time we add an arc that passes through the radar range of a SAM we must determine the true amount of time the aircraft is within range of the SAM in order to determine the risk units for the path. In the path-independent model, re-computing is not required. Additionally, the path-dependent model may require more enumeration since the objective function is non-linear.

Figures 3 and 4 show sample output results, demonstrating the differing paths chosen. Optimal routes in the path-independent model (Figure 3), seek to reduce the amount of time in the inner most SAM range circles. In these areas, the probability of shooting an aircraft down is higher than the outer annulus shaped regions. In the path-dependent model (Figure 4), optimal routes seek to reduce the amount of time spent within range of a SAM, regardless of the route location within the range circle.

Since the path-independent model always generates routes in less run time, it would be preferred whenever possible over the path-dependent model. In environments where the path-dependent model better suits the tactical situation, longer run times would be acceptable in order to find the optimal route. With the methods outlined in this thesis, air intelligence analysts can now make the appropriate modeling decisions.

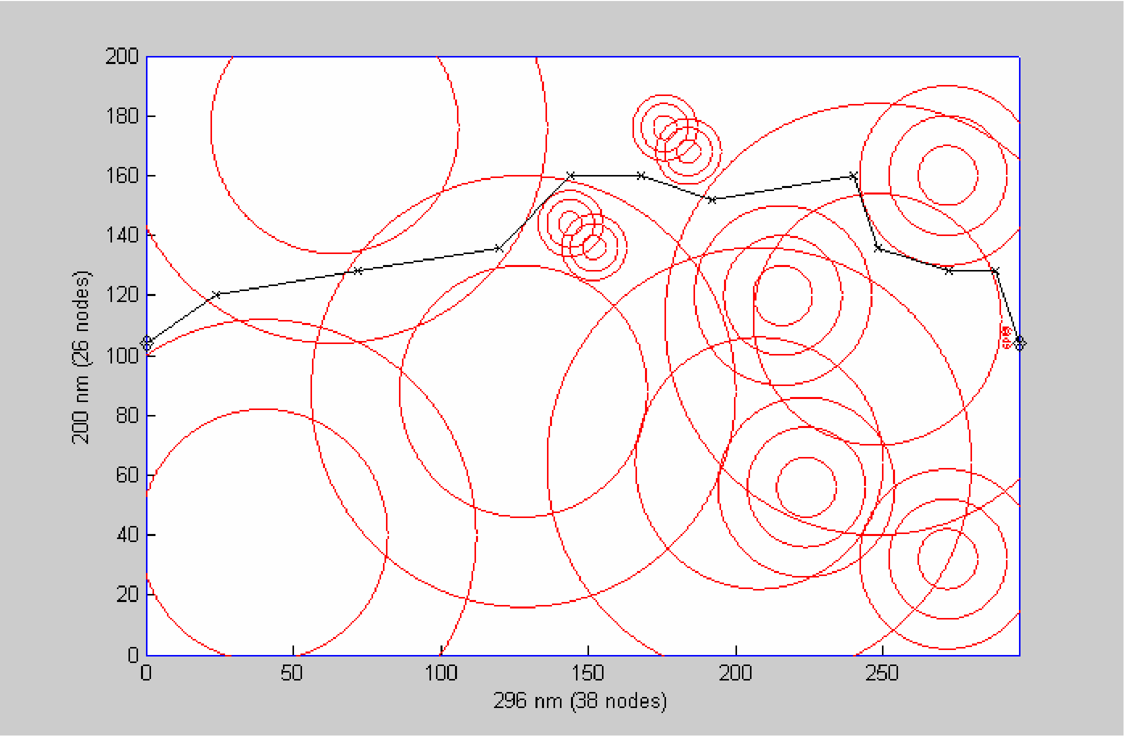

Figure 3. Route from the path-independent model constrained by 350 units of fuel.

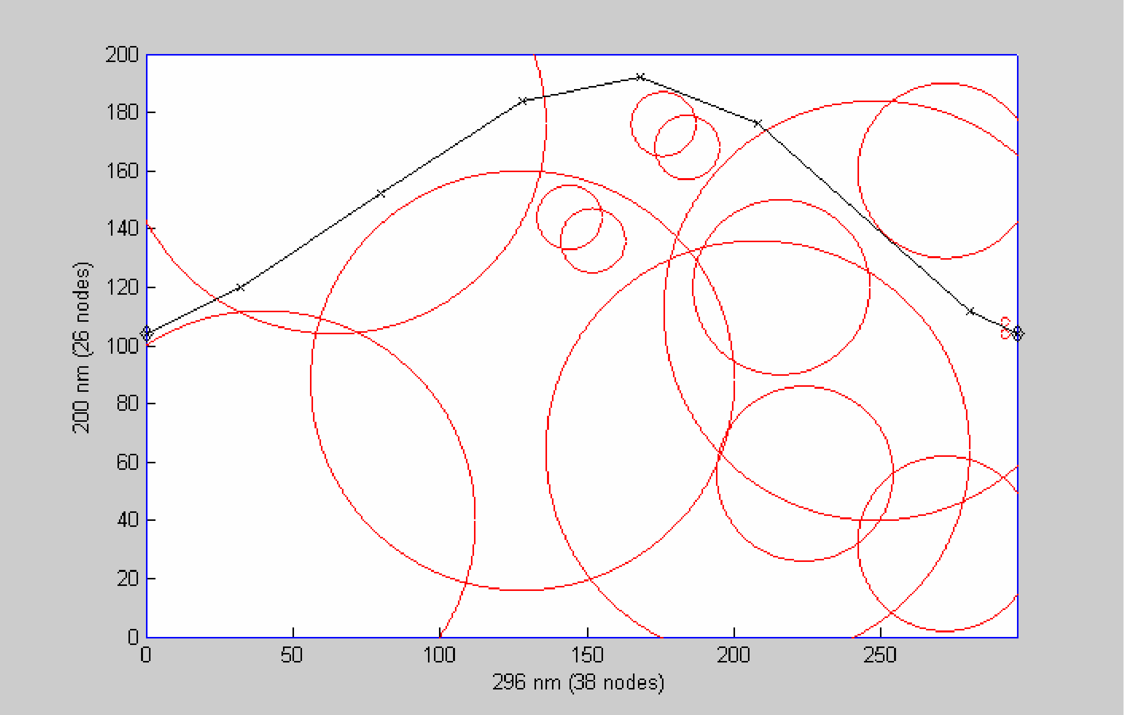

Figure 4. Route from path-dependent model constrained by 350 units of fuel.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. MULTIPLY-CONSTRAINED THREE-DIMENSIONAL SCENARIO

In this chapter, we generate a three-dimensional aircraft route from origin to destination in the presence of multiple radar threats and locations where the enemy may observe and react to the flight.

A. SCENARIO

1. Terrain

The area of interest is a plateau with an elevation of 295 feet cut by a large canyon whose floor has an elevation of zero feet (see Figure 4). The area is 120 kilometers from east to west, by 90 kilometers from north to south. The aircraft seeks to travel from the southwest corner at an altitude of 580 feet to the northeast corner at a final altitude of 580 feet.

2. Sources of Risk

There are two radars (depicted with the white stars in Figure 4) which have a 10 percent per minute probability of detecting and destroying the aircraft if it is within their 59.8 kilometers range and the line of sight is unobstructed by terrain. These radars are located four kilometers north, 88 kilometers east and 60 kilometers north, 88 kilometers east of the southwestern most portion of the airspace. Additionally there are three locations (depicted with gray stars) where enemy are suspected to be located. They are located 12 kilometers north and 38 kilometers east, 49 kilometers north and 62 kilometers east, and 73 kilometers north and 114 kilometers east of the southwestern most corner of the airspace. The first two have a maximum observation range of ten kilometers and the last has a maximum observation range of seven kilometers, provided they have a clear line of sight unobstructed by terrain. The enemy will shoot down aircraft entering these detection ranges with a 70 percent probability, if they remain within, or enter another of these observation areas after ten minutes of entry or within sixty minutes of departure from one of these areas. Finally, there is a risk per kilometer, p_{alt}, which decays as you climb, representing chance of mission failure due to striking an obstacle or encountering small arms fire. Let H_s, H_e, and U represent the starting altitude, ending altitude and distance traveled on the segment of flight represented by an arc, the risk from this source of threats on this arc is then given by the equation:

$$p_{alt} = 5 \cdot 10^{-5} U \left[\exp\left(-H_s\right) + \exp\left(-H_e\right) \right] \tag{19}$$

This risk is in addition to the risks mentioned previously.

3. Airspace

The airspace is broken into a lattice of potential waypoints to which the aircraft may fly. The lattice has four possible altitudes in increments of 145 feet, beginning at 145 feet. The horizontal lattice is spaced at one-kilometer intervals. A lattice point must be at least 50 feet above the terrain in order to be a waypoint to which the aircraft can fly. These lattice points are the nodes in our network.

We generate the arcs in our network by connecting a lattice point with 24 other lattice points. We connect a given lattice point with points in eight directions on the same altitude level and with the 16 directly above and below those eight. Which lattice points we actually connect depends upon the arc length size we allow in the north/south and east/west component directions.

4. Aircraft

The aircraft travels at a constant ground speed of 160 kilometers per hour. It consumes fuel at rates of 20 pounds per kilometer when in a climb, 12 pounds per kilometer in descents, and 15 pounds per kilometer when in level flight.

B. RESULTS

1. Initial Results

We carried out our computations on a Dell Precision PWS690 Intel® Xeon™ CPU 3.37GHz processor, with 3.00 GB of RAM, with programs written and compiled using Microsoft Visual C++ Version 6.0. The minimum time required to fly direct from the origin to the destination is 59 minutes. The minimum fuel required to fly from the origin to the destination is 2359 pounds. In Table 2, we consider side constraint limits fixed at a flight time limit of 75 minutes and fuel capacity of 3000 pounds. We set the optimality tolerance at two percent.

Case	Flight Segment Length		Number of Arcs in Route	Run Time (sec)
	East/ West (km)	North/ South (km)		
1	12	15	11	0.031
2	12	9	14	0.031
3	10	10	15	0.032
4	10	9	17	0.469
5	8	9	20	0.547
6	8	6	24	0.766
7	6	6	29	0.984
8	5	5	35	9.047
9	4	5	40	378.29
10	3	5	45	1.563

Table 2. Run times with varying flight segment lengths. Fuel Capacity 300 lbs and Flight Time Limit of 75 minutes

In Table 2, run time grows as the flight segment length decreases. As the length of flight segment decreases the number of potential paths to enumerate increases. Additionally, the number of arcs in a path from the origin to the destination increases as the flight segment length decreases. Case number nine had the longest run time of the cases listed in Table 2. Decreasing the segment flight length further proved impractical as the run times exceeded 24 hours.

Figures 5 and 6 depict the overhead and horizontal views of the route of flight when the aircraft flight length is three kilometers in the east and west directions and five kilometers in the north and south directions. The route hugs the lower edge of the airspace before proceeding north and dropping in to the canyon below the line of sight of the surface radars. The route also flies directly over a suspected enemy location; however, the aircraft enters and exits their range before they have sufficient time to react.

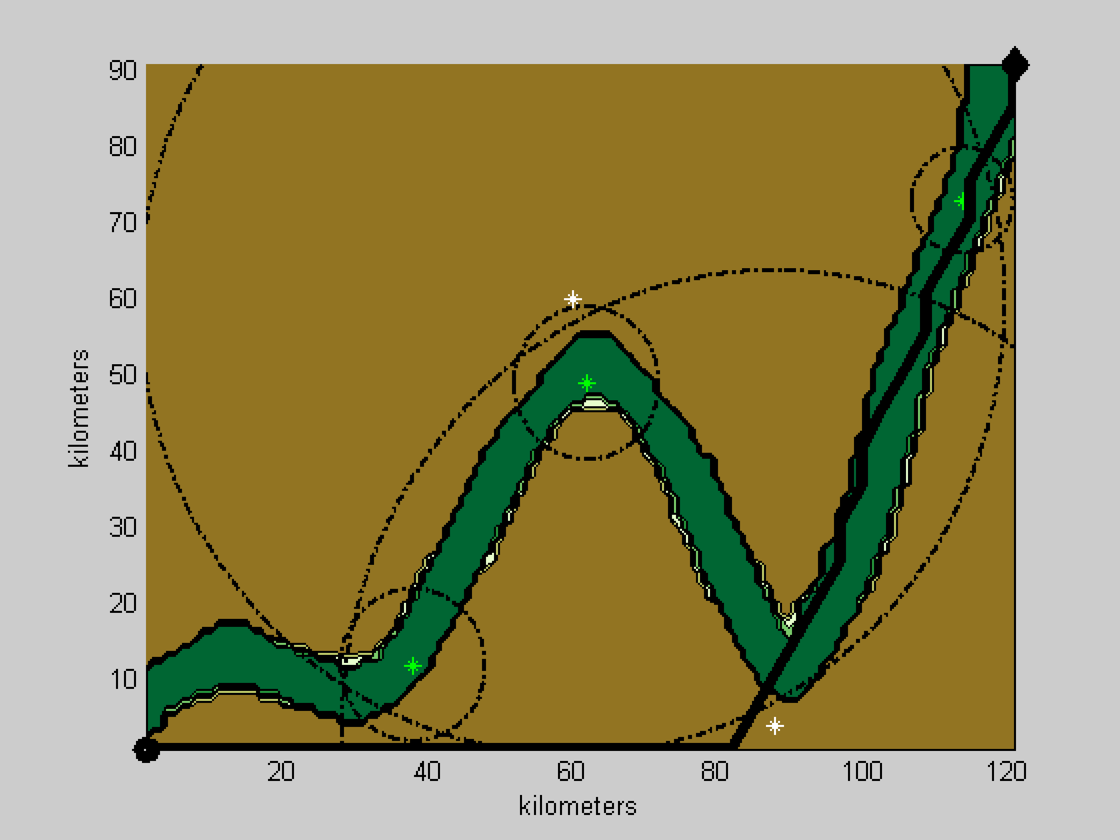

Figure 5. Overhead view of route when fuel capacity is 3000 lbs and flight time limit is 75 minutes. The flight segment length is 3 km in the east/west direction and 5 km in the north/south direction.

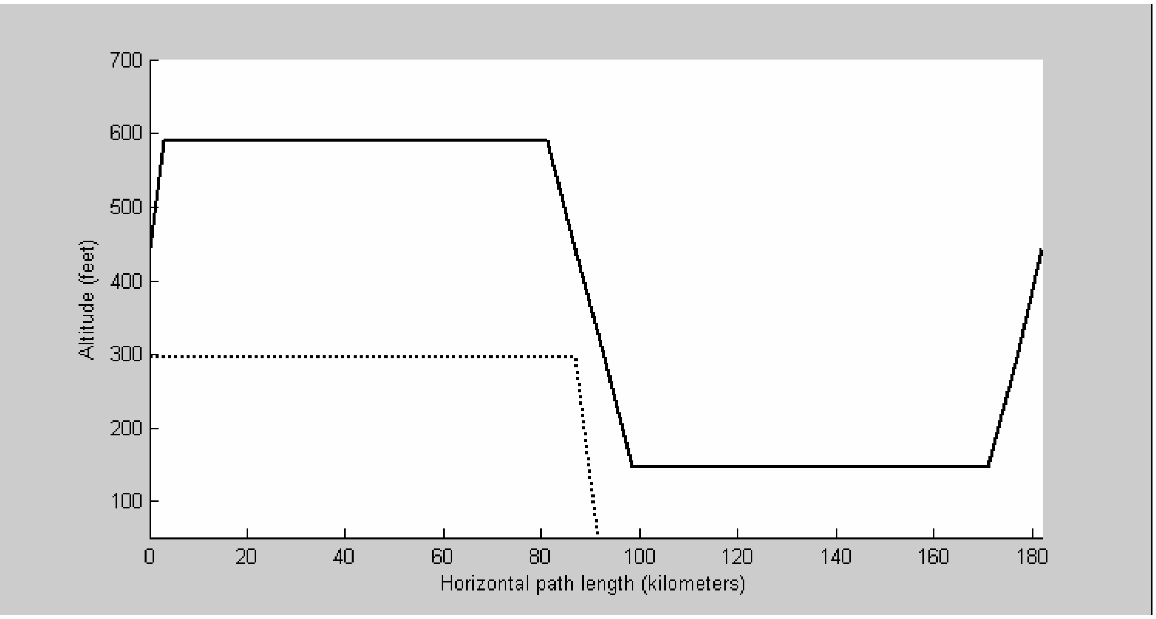

Figure 6. Horizontal view of route when fuel capacity is 3000 lbs and flight time limit is 75 minutes. The solid line represents the aircraft altitude along the route. The dotted line represents the corresponding terrain altitude along the route.

Varying the values of the side constraints also has an effect on the run times. Table 3 lists the run time results in seconds for a network that allowed flight segment lengths of five kilometers in any of the four cardinal directions while the side constraints varied as listed. We set the optimality tolerance at two percent.

Fuel Capacity (lbs)	Flight Time Limit (min.)						
	60	65	70	75	80	85	90
2400	0.47	0.47	0.45	0.47	0.45	0.45	0.45
2500	0.05	0.34	0.38	0.38	0.36	0.38	0.39
2600	0.05	13.98	14.35	14.41	14.41	14.39	14.47
2700	0.05	30.27	49.25	51.23	51.11	51.31	51.44
2800	0.05	62.02	73.44	90.91	114.11	127.31	101.74
2900	0.05	193.22	334.14	33.55	38.61	34.52	35.52
3000	0.05	18.61	1613.49	9.05	9.89	8.94	9.22
3100	0.05	18.66	6053.63	49.16	57.89	40.88	41.45

Table 3. Run times in seconds as fuel capacity and flight time limit vary

As the both resources increase in proportion, the run time generally increases. There are, however, certain combinations of constraints resulting in faster run times. For almost all fuel constraints, a flight time limit of 70 minutes resulted in the longest run time. The run times for this mission time limit with fuel resources of 3000 pounds or more are too long for realistic use in mission planning. However, modification of the creation of the network in the manner outlined in the next section, result in faster run times acceptable for employment in mission planning.

2. Results Following Network Expansion

In order to improve the method run time, we perform a network expansion. We create a new directed network $G' = (N', E')$, where N' is the expanded set of nodes and E' is the expanded set of arcs. In order to accomplish this expansion, we generate multiple copies, referred to as levels, of the network formed in the original fashion. Each copy represents the time since flight over a location of enemy.

The ten-minute reaction time of the enemy and the minimum amount of flight time along an arc determine the number of levels required. Let $f^1_{(i,j)}$ correspond to the flight time for an aircraft along arc $(i,j) \in E$. We generate a set $\Omega = \{0,10\} \cup \left\{ f^1_{(i,j)} \mid (i,j) \in E, f^1_{(i,j)} < 10 \right\}$. Now for all pairs of elements $\omega_1, \omega_2 \in \Omega$, if $\omega_3 = \omega_1 + \omega_2 < 10$, then replace Ω with $\Omega \cup \{\omega_3\}$. As you augment Ω with a new element, you repeat this process until no further augmentation is possible. The total number of levels required will increase as the minimum amount of time required to fly along an arc decreases.

We generate the nodes in our expanded network $\langle \omega, i \rangle \in N'$, for all $\omega \in \Omega$ and $i \in N$. The origin node in our network is $\langle 0, s \rangle$ and our destination node is $\langle 0, t \rangle$. We generate the arcs in our expanded network as $(\langle \omega_1, i \rangle, \langle \omega_2, j \rangle) \in E'$. The arcs in the expanded network relate to the arcs in the original network in the following ways:

1. At level $\omega = 0$, for $(i, j) \in E$, if node i is not a location of enemy, the corresponding arc is $(\langle 0,i\rangle, \langle 0, j\rangle) \in E'$.

2. At level $\omega = 0$, for $(i, j) \in E$, if node i is a location of enemy, the corresponding arc is $(\langle 0,i\rangle, \langle \omega_1, j\rangle) \in E'$. Where $\omega_1 = \min\left\{ f^1_{(i,j)}, 10\right\}$.

3. At all levels $0 < \omega_1 < 10$, for $(i, j) \in E$, if $j \neq t$, the corresponding arc is $(\langle \omega_1,i\rangle, \langle \omega_2, j\rangle) \in E'$. $\omega_2 = \min\left\{\omega_1 + f^1_{(i,j)}, 10\right\}$.

4. At level $\omega = 10$, for $(i, j) \in E$, if $j \neq t$, the corresponding arc is $(\langle 10,i\rangle, \langle 10, j\rangle) \in E'$.

5. At all levels $\omega > 0$, for $(i,t) \in E$, the corresponding arc is $(\langle \omega,i\rangle, \langle 0,t\rangle) \in E'$.

This process makes intuitive sense because each level represents the time since the aircraft last flew over the enemy. Level $\omega = 0$ indicates prior flight has avoided enemy locations, and level $\omega = 10$ indicates it has been ten minutes or longer since flight over a location of enemy.

For an arc $(\langle \omega_1,i\rangle, \langle \omega_2, j\rangle) \in E'$ in the expanded network, the primary data, $c'_{(\langle \omega_1,i\rangle,\langle \omega_2,j\rangle)}$, and secondary data, $f'^k_{(\langle \omega_1,i\rangle,\langle \omega_2,j\rangle)}$ for all $1 \le k \le K$, are related to the primary data and secondary data of the original network by the following relations:

1. $f'^k_{(\langle \omega_1,i\rangle,\langle \omega_2,j\rangle)} = f^k_{(i,j)}$ for all $1 \le k \le K$.

2. $c'_{(\langle \omega_1,i\rangle,\langle \omega_2,j\rangle)} = c_{(i,j)}$ for all $\omega_1, \omega_2 < 10$.

3. $c'_{(\langle \omega_1,i\rangle,\langle 10,j\rangle)} = c_{(i,j)} - \ln(0.3)$.

The third relation is the key to improvements in run time resulting from expansion. Since, flight over an enemy location on level $\omega = 10$ will always result in a 70 percent chance of this enemy shooting down the aircraft. This “risk unit” from the enemy, $-\ln(0.3)$, is in addition to other threats the aircraft might encounter from radar, altitude and small arms fire, $c_{(i,j)}$. In the network prior to expansion, $c_{(i,j)}$ corresponds to the path-independent probability of mission success obtained from relaxation of the PCSP in order to formulate the CSPP. At level $\omega = 10$, this relaxation is no longer required. In this scenario, constraints of time and fuel on the route preclude reaching the 60-minute time limit for threat from the enemy locations to "reset" due to lack of vigilance. If sufficient time and fuel allows reaching the 60-minute time limit, further expansion of the network is required for appropriate modeling.

In expanding the network that allowed flight segment lengths of five kilometers in any of the four cardinal directions we require 16 levels. In the original network $|N| = 25111$ and $|E| = 16156$. After expansion both the corresponding number of nodes and arcs increase by a factor of 16 to $|N'| = 401776$ and $|E'| = 258496$.

Table 4 lists the run time results in seconds after expansion of the network used to generate the data in Table 3. We set the optimality tolerance at two percent.

Fuel Limit (lbs)	Flight Time Limit (min.)						
	60	65	70	75	80	85	90
2400	0.78	0.80	0.80	0.86	0.83	0.81	0.80
2500	0.80	1.05	1.05	1.13	1.16	1.08	1.09
2600	0.78	1.20	1.22	1.28	1.30	1.25	1.23
2700	1.38	15.03	16.18	9.88	9.83	9.70	9.83
2800	0.80	15.34	1.14	1.19	1.22	2.95	2.97
2900	0.80	15.36	23.19	24.91	24.76	14.30	14.34
3000	0.81	9.63	23.19	25.23	25.00	14.65	14.58
3100	0.81	9.64	23.27	25.16	25.14	14.56	14.73

Some run time values have increased slightly due to the increased network size and corresponding increase in computing time for shortest-path calculations, but now all run times are low enough for employment of the method in a real-world mission planning environment. The most dramatic reduction in run times occurs for the flight time limit of

70 minutes with fuel capacities of 3000 pounds and 3100 pounds. These run times

decreased from 1613 seconds and 6054 seconds to 23.19 seconds and 23.27 seconds, respectively.

These reductions in run times result from an increase in the lower bound obtained from the LRSPP, $\underline{z}^*$. Table 5 lists the percentage of improvement in the lower bound obtained by the network expansion.

Fuel Limit (lbs)	Flight Time Limit (min.)						
	60	65	70	75	80	85	90
2400	6.02%	6.02%	6.02%	6.02%	6.02%	6.02%	6.02%
2500	9.27%	25.32%	25.32%	25.32%	25.32%	25.32%	25.32%
2600	9.27%	33.84%	33.76%	33.76%	33.76%	33.76%	33.76%
2700	9.27%	38.19%	38.85%	38.76%	38.76%	38.76%	38.76%
2800	9.27%	46.05%	9.37%	9.41%	9.46%	10.96%	9.35%
2900	9.27%	55.16%	23.45%	30.77%	30.96%	30.66%	30.66%
3000	9.27%	32.90%	38.59%	49.26%	49.63%	49.24%	49.24%
3100	9.27%	32.90%	69.56%	79.16%	79.94%	79.50%	79.56%

Table 5. Percentage improvement in lower bound obtained through improved network creation

We improved the lower bound a minimum of 6.02 percent in all cases. The greatest improvement of 79.94 percent occurred with a flight time limit of 80 minutes with a fuel capacity of 3100 pounds.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

A. CONCLUSIONS

In this thesis, we have built upon the algorithm developed by Carlyle et al. (2007b) as the basis for a method of automatically generating routes for aircraft, taking into account parameters that depend upon the route itself. This method can provide optimal routes in computational times fast enough for employment in tactical mission planning. Furthermore, the method is flexible enough to take into account terrain avoidance, line-of-site calculations for radar and visual detection, and varying aircraft speed and fuel consumption.

A key point revealed in the comparison of F/A-18 strike models is the differences obtained in modeling radar threats in a path-dependent or independent manner. Large differences in route selection occurred in the differing models. Radar models should be re-examined to determine if previous assumptions of path-independence are valid.

As an added benefit, an automatic route generator would enable an Air Mission Commander to perform limited sensitivity analysis of his mission and determine if significant improvements in the chance for mission success would occur from requesting additional resources. With data gleaned through this rapid analysis, the Air Mission Commander could quantify the benefit of any additional resources requested from higher headquarters.

Routes generated by the method developed in this thesis can have a greater chance of mission success than those manually generated or those that do not take into account path-dependent probabilities. The method run time is sensitive to the threat environment but generates routes sufficiently fast, provided the appropriate minimum distance for flight segment length is established. There does remain, however, room for expansion and refinement in order to provide a workable tool to the fleet. We describe such extensions next.

B. RECOMMENDATIONS FOR FUTURE RESEARCH

1. Radar Parameters and Dependency Function

In the F/A strike model, we chose a dependency function and then assigned radar parameters to ensure that an aircraft flying directly into the center of the radar would encounter the same amount of risk as in the path-independent model. Air intelligence analysts can now build upon the work of this thesis, developing and validating path-dependency models for radar threats.

In addition, the radar models utilized a line of sight determination in assigning probabilities, but we did not take environmental factors that might cause attenuation of detection probability based upon meteorological conditions into account. Future studies might further enhance the radar models in such a fashion.

2. Illumination Dependencies

In this thesis, we did not explore a scenario in which a threat to the aircraft resulted from changing illumination along the route. Future studies can develop this scenario by integrating meteorological illumination models for illumination along the route of flight. Such a tool would be useful to planners even in training environments to minimize risk when training aviators in the use of night vision devices in flight.

3. Previously Flown Routes

We only considered the route from an origin to a destination in this thesis. We also assumed that there had been no prior aircraft flights in the area. Over time, the enemy might base their actions on a pattern of aircraft activity. Future studies can incorporate a higher unit of risk for segments of flight previously flown. Such a dependency model would better enable a flight planner to plan not only his egress route as well, but also enable planners to utilize information of previous air activity in order to balance the threats from predictability, enemy and environment.

LIST OF REFERENCES

Ahuja, R. J., Magnanti, T. L., & Orlin, J. B. (1993). *Network Flows*, Englewood Cliffs, New Jersey: Prentice Hall.

Aneja, Y.P., Aggarwal, V., & Nair, K. P. K. (1983). Shortest Chain Subject to Side Constraints, *Networks*, 13, 295-302.

Carlyle, W. M., Royset, J. O., & Wood, R. K. (2007a). Routing Military Aircraft with a Constrained Shortest-Path Algorithm, in review.

Carlyle, W. M., Royset, J. O., & Wood, R. K. (2007b). Lagrangian Relaxation and Enumeration for Solving Constrained Shortest-Path Problems, in review.

CLOAR (2007). Common Low Observable Autorouter, BAE Systems, Battle Management Systems Gorup, San Diego, California. http://www.baesystems.com (accessed 8 September 2007)

Davies, C. & Lingras, P. (2003). Genetic Algorithms for Rerouting Shortest Paths in Dynamic and Stochastic Networks. *European Journal of Operations Research*, 144, 27-38.

Dumitrescu, I. & Boland, N. (2003). Improved Preprocessing, Labeling and Scaling Algorithms for the Weight-Constrained Shortest Path Problem. *Networks*, 42, 135-153.

FalconView (2007). Georgia Tech Research Institute, Atlanta, Georgia. http://www.falconview.org/default.htm (accessed 8 September 2007).

Fan, Y. Y., Kalaba, R. E., & Moore, J. E. III (2005). Shortest Paths in Stochastic Networks with Correlated Link Costs. *Computers and Mathematics with Applications*, 49, 1549-1564.

FM 90-26 (1990). *Airborne Operations*. Fort Monroe, Virginia: Army Training and Doctrine Command.

Inanc, T., Misovec, K., & Murray, R. M. (2004). Nonlinear Trajectory Generation for Unmanned Air Vehicles with Multiple Radars. *43rd IEEE Conference on Decision and Control Proceedings*, 4, 3817-3822.

Milam, M.B. (2003). *Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems*. Ph.D. Dissertation, California Institute of Technology, Pasadena, California.

Misovec, K., Inanc, T., Wohletz, J., & Murray, R.M. (2003). Low-Observable Nonlinear Trajectory Generation for Unmanned Air Vehicles. *42nd IEEE Conference on Decision and Control Proceedings*, 3103-3110.

NPSOL 5.0 (2007). Stanford Business Software, Inc., Palo Alto, California. http://www.sbsi-sol-optimize.com/asp/sol_product_npsol.htm (accessed 8 September 2007).

OPUS (2007). OR Concepts Applied, Whittier, California. http://www.orconceptsapplied.com (accessed 8 September 2007).

Orda, A. & Rom, R. (1990). Shortest Path and Minimum-Delay Algorithms in Networks with Time-Dependent Edge-Length. *Journal of the Association for Computing Machinery*, 37, 607-625.

Tan, J. & Leong, H. W. (2004). Least-Cost Path in Public Transportation Systems with Fare Rebates that are Path- and Time- Dependent. *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 1000-1005.

Tharp, J. (2003). JRAPS for Mission Planning. *Northrop Grumman Information Technology*, TASC, Sterling, Virginia.

Zabarankin, M., Uryasev, S., & Murphey, R. (2006). Aircraft Routing under the Risk of Detection. *Naval Research Logistics*, 53, 728-747.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California

7. Director, Studies and Analysis Division, MCCDC, Code C45
   Quantico, Virginia

8. Assistant Professor Johannes O. Royset
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California

9. Assistant Professor Ralucca M. Gera
   Department of Applied Mathematics
   Naval Postgraduate School
   Monterey, California

10. Associate Professor Craig W. Rasmussen
    Department of Applied Mathematics
    Naval Postgraduate School
    Monterey, California